

Inicio | Fundamentos | Instalación | Añadir Elementos | Encadenar Métodos | Asociar Datos | Usando sus Datos | Desplegar DIVs | La Función data( ) | Introducción a SVG | Despliegue de SVG | Tipos de Datos | Diagrama de Barras | Diagrama de Dispersión | Escalas | Ejes |

## Desplegar DIVs

Es hora de comenzar a crear diagramas con datos.

Por última vez vamos a utilizar el mismo conjunto de datos.

```
var dataset = [ 5, 10, 15, 20, 25 ];
```

Este conjunto se va a usar para generar un diagrama de barras muy sencillo. Los diagramas de barras son en esencia rectángulos y la forma más fácil de dibujar un rectángulo es usando la etiqueta de HTML `<div>`. (En realidad para los navegadores, *todo* es un rectángulo, entonces es posible cambiar el ejemplo y usar `<span>` o el elemento que prefiera).

El siguiente `div` funciona bien como una barra de datos.



```
<div style="display: inline-block;  
  width: 20px;  
  height: 75px;  
  background-color: teal;"></div>
```

(Este ejemplo muestra algo que no se debe hacer. Usualmente, no se debe usar un `div` vacío para efectos visuales únicamente, sin embargo para efectos de este tutorial se puede hacer la excepción.)

Puesto que este es un `div`, su ancho `width` y altura `height` se definen en los estilos de CSS. Cada barra en el diagrama puede compartir las mismas propiedades de despliegue (con excepción de altura `height`), por consiguiente se pueden incluir los estilos compartidos dentro de una clase `class` de nombre `bar`:

```
div.bar {  
  display: inline-block;
```

```
width: 20px;  
height: 75px; /* Esto se cambiará próximamente */  
background-color: teal;  
}
```

Ahora a cada `div` se le debe asignar la clase `bar`, de tal manera que la regla de CSS se aplique. Si estuviera escribiendo código HTML a mano, escribiría:

```
<div class="bar"></div>
```

Con D3, para añadir una clase a un elemento se utiliza el método `selection.attr()`. Es importante entender la diferencia entre `attr()` y su pariente cercano, `style`.

## Asignación de Atributos

`attr()` se usa para asignarle un atributo y su valor a un elemento. Un atributo de HTML es cualquier dupla propiedad/valor que se pueda insertar dentro de los corchetes. Por ejemplo, estos elementos de HTML

```
<p class="caption">  
<select id="country">  

```

contienen en total cinco atributos (con sus valores correspondientes), y todos pueden ser asignados con la función `attr()`:

```
class | caption  
id    | country  
src   | logo.png  
width | 100px  
alt   | Logo
```

Para asignarle la clase `bar` a los `divs`, se puede usar:

```
.attr("class", "bar")
```

## Una Nota sobre Clases

Tome nota de que la clase de un elemento se guarda como un atributo de HTML. La clase, a su vez, se usa para referenciar la regla del estilo de CSS. Esto puede causar algo de confusión porque existe una diferencia entre asignar una clase (de la cual se infieren los estilos) y aplicar un estilo directamente a un elemento. En D3 es factible hacer

ambos, y si bien se puede utilizar el que considere más conveniente, yo recomiendo usar *clases* para aquellas propiedades que sean compartidas por varios elementos, y aplicar reglas de *estilo* directamente solo cuando se esté desviando de la norma (que es exactamente lo que haremos en un momento.)

Vale la pena mencionar otro método de D3, `classed()`, que puede ser usado para aplicar o remover clases rápidamente de los elementos. La línea de código anterior pudo haber sido escrita así: `.classed("bar", true)`


## Otra Vez con las Barras

Ahora, se puede ver todo el código de D3 hasta el momento, incluyendo el conjunto de datos:

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body").selectAll("div")
  .data(dataset)
  .enter()
  .append("div")
  .attr("class", "bar");
```





Puede ver [esta página con código](#). No olvide revisar el código fuente y abrir el inspector de web para ver qué está pasando. Debe ver cinco barras verticales, cada una representando a un dato del conjunto. Como no hay espacio entre las barras, la gráfica aparece como un solo rectángulo.

## Definición de Estilos

El método `estilo()` se usa para aplicar la propiedad y valor de CSS a un elemento de HTML. Esto es el equivalente a añadir reglas de CSS dentro del atributo `style` en el código HTML, así:

```
<div style="height: 75px;"></div>
```

En el diagrama de barras, la altura de cada barra debe generarse de acuerdo con el dato correspondiente, entonces si se añade el siguiente código:

```
.style("height", function(d) {  
    return d + "px";
```

```
});
```



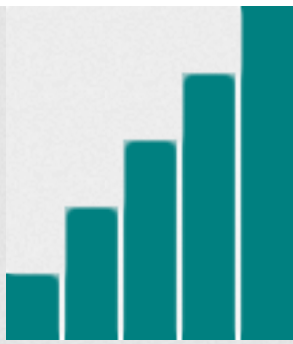
verá un diagrama de barras muy pequeño. [página de ejemplo](#).

Cuando D3 pasa por cada dato, asocia ese dato a `d`, por consiguiente se está asociando un valor `d` al atributo `height` y se le añade `px` (para especificar las unidades en pixeles). El resultado son alturas de 5px, 10px, 15px, 20px, y 25px.

Esto suena un tanto absurdo, entonces hagamos las barras más altas:

```
.style("height", function(d) {  
  var barHeight = d * 5; //Incrementar la escala 5 veces.  
  return barHeight + "px";  
});
```

e incluyamos un espacio en el lado derecho de cada barra, para alinear todo mejor.



Excelente! Ya podemos ir a SIGGRAPH con este diagrama.

Acá está la **página final** de ejemplo con este código. Nuevamente, es importante revisar el código y utilizar el inspector de web para comparar el HTML original con el DOM final.

**Siguiente: El Poder de la Función data( ) →**

---

Este tutorial fue traducido por Gabriel Coch

Todo el contenido fue desarrollado por y le pertenece a Scott Murray